

# SS法による量子多体系の大次元対角化計算

水崎 高浩

## 概要

これまで量子多体系の大規模な厳密対角化計算は主として Lanczos 法およびその改良された方法で行われてきたが、最近 SAKURAI と SUGIURA によって SS 法という複素積分を用いた非常に有望な方法が提案された。これら 2 つの方法は、クリロフ部分空間を利用する点で類似点はあるが、理論形式は全く異なるものである。本稿では、2 つの方法の差異および超大次元計算における可能性について考察し、いくつかの収束性の改善について議論する。

## 1 はじめに

量子多粒子系の解法の中で、厳密対角化法は、適用できる系が小さいという問題点を除いては、様々な量が厳密に計算できるため、計算可能な範囲で何らかの結果や示唆が得られる場合には非常に役立つ。また、昨今の PC 環境の向上や並列計算機の発展により、適用できる系のサイズも格段に大きくなってきている。

厳密対角化法は、これまで Lanczos 法 [1] およびその改良された方法で行われてきており、近年、理論的に大きな発展はなかったが、2003 年に SAKURAI と SUGIURA によって複素関数論を使った新しい方法が提案された [2]。これは Lanczos 法とは全く異なる新しい方法であるので、量子系の超大次元計算に適するかどうか、また、その特性をよく研究する必要がある。そのため、SS 法を使ったシュル模型計算の可能性について、論文 [3] で報告した。本稿では、それ以降の発展を超大次元に適した対角化法や励起状態に関する収束性およびその改良についてまとめる。

## 2 Lanczos 法と SS 法

量子物理での計算を考えるために、以降、量子力学での記法を用いる。

## 2.1 Lanczos 法

Lanczos 法 [1] は、対称行列  $A$  の最小固有値または最大固有値およびそれに近い幾つかの固有値、固有ベクトルを効率よく求める方法である。初期ベクトル  $x_0$  に対して、行列  $A$  を何度も掛けて作られるクリロフ (Krylov) 部分空間、すなわち、 $\{x_0, Ax_0, A^2x_0, \dots\}$  により真の固有値を近似する方法である。簡単にこの部分空間の性質を知るために、 $x_0$  を  $A$  の完全系で展開する。すなわち、 $x_0 = \sum c_i x_i$ 。ただし、 $c_i$  は係数で  $x_i$  は  $A$  の固有ベクトルであり、対応する固有値を  $e_i$  とする。 $A$  をかけることで  $A^p x_0 = \sum c_i e_i^p x_i$  となることから、 $A^p x_0$  ベクトルの中の固有値の小さい成分は  $p$  が大きくなるにつれ小さくなる。したがって、超大次元の固有値問題を、クリロフ部分空間の次元の関数としてその固有値の収束をみることで、極めて小さな次元（ほとんどの問題では、問題としている系の次元に関わらずほぼ数十から数百次元）の固有値問題に帰着させることができる。また、収束した固有値が求まれば、対応する固有ベクトルも求まる。

Lanczos 法で量子多粒子系の問題を解くと、まず基底状態のエネルギーが求まり、順次、基底状態近傍の励起状態のエネルギーが求まっていくというように、物理的に興味のある量子状態が効率よく求められるため、Lanczos 法はよく利用されてきている。実際の計算には、後述する Thick Restart Lanczos 法 [4] が便利である。

## 2.2 SS 法

次に、SAKURAI と SUGIURA によって発見された SS 法のアルゴリズム [2] を考えよう。これは量子物理の応用に限らず、非常に一般的な固有値問題の解法であり、特に非エルミート行列の固有値にも使えるのは興味深い点である。しかしながら、本稿ではエルミート行列に帰着される量子物理学の問題に限る。

SS 法で中心となるのは、次式で定義されるモーメント  $\mu_p$  ( $p = 0, 1, 2, \dots$ ) である。モーメントは整数  $p$  に対して、コーシー (Cauchy) 積分を用いて、

$$\mu_p = \frac{1}{2\pi i} \int_{\Gamma} \langle \psi | \frac{(z - \epsilon)^p}{z - H} | \phi \rangle dz \quad (1)$$

と定義される量である。ここで、 $|\psi\rangle$  と  $|\phi\rangle$  は求めたい波動関数と直交しない任意の波動関数であり、 $H$  はハミルトニアンである。 $\Gamma$  は積分路であり、求めたい固有値のまわりの閉曲線をとる。被積分関数の分母は  $z - H$  なので、 $H$  の固有値がこの積分の極となり、積分路  $\Gamma$  に囲まれるエネルギー固有値に対応する留数で積分値は表される。一方、この積分値は積分路  $\Gamma$  に沿った積分をすることでも評価できる。

このモーメントの積分値を留数積分で表すために、 $H$  の完全系  $|\varphi_k\rangle$  ( $k = 1, \dots, N$ ) を考える。 $|\varphi_k\rangle$  は、 $H|\varphi_k\rangle = e_k|\varphi_k\rangle$  を満たし、 $e_k$  は  $|\varphi_k\rangle$  に対する  $H$  の固有値とする。

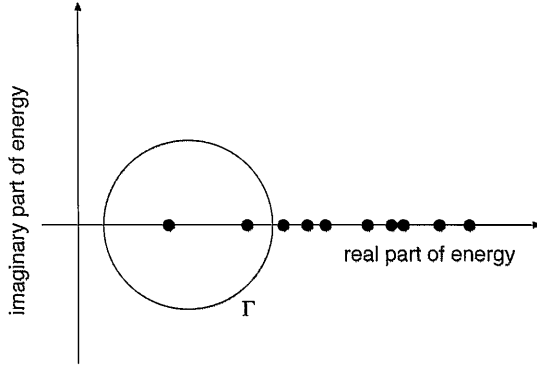


図 1: SS 法では、エネルギー固有値がポールとなる複素数値をとる被積分関数を考え、積分路  $\Gamma$  による複素積分を考察する。

$|\psi\rangle$  と  $|\phi\rangle$  をこの完全系で展開したときの  $|\varphi_k\rangle$  係数をそれぞれ  $c_k, d_k$  とすると、モーメントは

$$\mu_p = \sum_{k \in \Gamma} (e_k - \epsilon)^p c_k d_k \quad (2)$$

のように表すことができる。ただし、 $k$  の和は閉曲線  $\Gamma$  に含まれる固有値に対してとる。したがって、モーメント  $\mu_p$  から  $e_k$  が求められれば、固有値問題が解けたことになる。

モーメントから  $e_k$  を求めるのはやや技巧的で、SS 法ではモーメントから次の 2 つの Hankel 型の  $p \times p$  の行列  $M$  と  $N$  を  $M_{ij} = \mu_{i+j-1}$  と  $N_{ij} = \mu_{i+j-2}$  と定義し、 $p$  次元の一般化固有値問題

$$Mx = \lambda Nx, \quad (3)$$

を解くことに帰着させる [2]。実際の計算では、 $p$  は 1 ～ 10 程度である。また、この  $\lambda$  が  $(e_k - \epsilon)$  に対応する。この技巧的な方法は論文 [5] にみられる。

SS 法の重要な点は、モーメントを求めるための積分計算である。積分路  $\Gamma$  にはいろいろな取り方がありうるが、たとえば、 $z = \epsilon$  を中心とした円を、 $z = \epsilon + re^{i\theta}$  のようにとると数値計算は容易になる。 $\epsilon, r$  は実であり、 $\theta$  は 0 から  $2\pi$  をとる。ターゲットとなる固有値は、 $\epsilon - r$  から  $\epsilon + r$  の間のものになる。 $\theta$  に関する積分は、数値積分で行うと、

$$\mu_p \sim \frac{1}{N_0} \sum_{k=0}^{N_0-1} \langle \psi | \frac{(z_k - \epsilon)^{p+1}}{z_k - H} | \phi \rangle \quad (4)$$

のような積分点の和になる。ただし、 $z_k = \epsilon + re^{i\frac{2\pi}{N_0}(k+\frac{1}{2})}$ 。複素平面上の対称性やス

ケール変換を行うと計算しやすい [2, 3]。

この式の数値計算上の問題点は、 $H$  が分母にあることである。これは、論文 [2] によれば、 $|\chi\rangle$  を

$$|\phi\rangle = (z - H)|\chi\rangle \quad (5)$$

により定義し、 $|\chi\rangle = (z - H)^{-1}|\phi\rangle$  と求めることで解決する。実際に  $|\chi\rangle$  を求めるには、連立方程式を COCG 法によって解けばよい。COCG 法は、 $\{|\phi\rangle, (z - H)|\phi\rangle, (z - H)^2|\phi\rangle, \dots\}$  というクリロフ (Krylov) 部分空間を利用した解法であり、Lanczos 法が利用する部分空間と本質的に同等のものである。

しかしながら、各積分点  $z_k$  毎に、連立方程式を解く必要があり、このままでは、計算量が Lanczos 法よりも積分の分点数倍必要になってしまうという困難がある。この点は、次に述べるシフト COCG 法 [6] を用いることで解決でき、実質的に積分点 1 点に対して計算するだけでよい [7]。つまり、 $z = z_0$  で連立方程式  $|\phi\rangle = (z_0 - H)|\chi\rangle$  を解けば、 $z$  が  $z_0$  近傍に対する  $|\phi\rangle = (z - H)|\chi\rangle$  は、もともと計算時間のかかるハミルトニアン行列要素の生成や作用を扱うことなく計算できる。この点は SS 法の応用上もとても重要な点である [2, 3, 8]。この状況を同一積分路上のシフトに対して図 2 に示したが、シフトは異なる積分路に対しておこなっても構わないものである。以上により、SS 法でエネルギー固有値が求められる。また、同様に波動関数も求めることができる [2, 3, 8]。

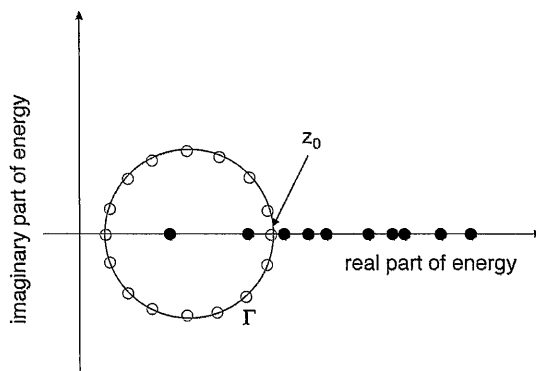


図 2: 複素積分を実行するためには、円周上に白丸で表したように積分点をとる、その各積分点で連立方程式を解く必要がある。しかし、実際の数値計算では図の  $z_0$  と書いた点でのみ連立方程式を解けばよく、その他の積分点における解はシフト法により容易に求めることができる。

Lanczos 法も SS 法も、クリロフ部分空間を使って固有値、固有ベクトルを求める方法である。クリロフ部分空間は、ハミルトニアン行列を初期波動関数に何度も掛けてできる部分空間なので、初期波動関数がよい対称性をもつように準備すれば、原理的には、生成されるクリロフ部分空間も同じ対称性をもつことになる。実際にこの性質は、 $M$  スキームのシュル模型計算では、角運動量保存を扱うのにこれまで使われてきた。Lanczos 法の場合は初期 Lanczos ベクトルを角運動量の固有状態にすれば、生成される Lanczos ベクトルも同じ角運動量をもつことになる。しかしながら、Lanczos 法のもつ数値的不安定性のため、Lanczos ベクトルの再直交化や  $J^2$  項をハミルトニアンに付加するというテクニックが必要となることがよく知られている。一方、SS 法の場合は数値的にかなり安定であり [3]、後述するように数値計算の負担がかなり軽くなる。

論文 [3] では、数値的に安定であるという事実を示しただけだったので、ここではその理由を考えてみよう。SS 法のモーメント  $\mu_p$  の中で  $\langle \psi | \frac{1}{z-H} | \phi \rangle$  の部分に着目する。 $|\psi\rangle$ 、 $|\phi\rangle$  は初期ベクトルであり、 $|\psi\rangle = |\phi\rangle$  ととると、 $|\chi\rangle = (z-H)^{-1}|\phi\rangle$  であるから、 $\langle \psi | \frac{1}{z-H} | \phi \rangle = \langle \phi | \chi \rangle$  になる。 $|\phi\rangle$  を角運動量のよい状態に選ぶ場合、角運動量射影演算子を  $P_J$  と書くと、 $|\phi\rangle = P_J|\phi\rangle$  と書いても同値である。 $|\chi\rangle$  は COCG 法により多くの行列ベクトル演算で求められるので、場合によっては、数値誤差のために、角運動量をすこし壊すかもしれないが、その場合も  $\langle \psi | \chi \rangle = \langle \phi | P_J | \chi \rangle$  なので、モーメントの計算自体は、角運動量を保つことができる。もちろん、数値誤差が致命的な場合で COCG 法自体の精度がない場合はこの限りではないが、SS 法が Lanczos 法に比べて数値的に安定な一つの原因は、このメカニズムのためだと考えられる。

## 2.3 シフト COCG 法

連立方程式の数値的解法というのは、物理というよりも応用数理に属する問題であるが、SS 法を実用的にするためには、その理解は必須である。SS 法に現れる連立方程式を解くための複素直交共役傾斜 (complex orthogonal conjugate gradient, COCG) 法は、CG 法の複素数で対称であるがエルミートではない場合への拡張である。 $(z-H)$  の行列表示を  $A$  とする。

$Az = b$  を解くために、3つのベクトル  $x_k$ ,  $r_k$ ,  $p_k$  を

$$x_{k+1} = x_k + \alpha_k p_k, \quad (6)$$

$$r_{k+1} = r_k - \alpha_k A p_k, \quad (7)$$

$$p_{k+1} = r_{k+1} + \beta_k p_k, \quad (8)$$

の漸化式で定義する [6]。ただし  $\alpha_k = r_k^T r_k / p_k^T A p_k$ ,  $\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$  である。

次に、対角成分をシフトした方程式  $(h - \sigma I)x^\sigma = b$  を考えると、この  $k$  番目の残差ベクトル  $r_k^\sigma$  は、 $\sigma = 0$  の連立方程式の  $k$  番目の残差ベクトル  $r_k$  と

$$r_k^\sigma = \frac{1}{\pi_k^\sigma} r_k \quad (9)$$

という関係で結びついている [7, 9]。  $\pi_k^\sigma$  は、

$$\pi_{k+1}^\sigma = (1 + \alpha_k \sigma) \pi_k^\sigma + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} (\pi_k^\sigma - \pi_{k-1}^\sigma), \quad (10)$$

$$\alpha_k^\sigma = \frac{\pi_k^\sigma}{\pi_{k+1}^\sigma} \alpha_k, \quad (11)$$

$$\beta_k^\sigma = \left( \frac{\pi_k^\sigma}{\pi_{k+1}^\sigma} \right) \beta_k \quad (12)$$

という漸化式で決まる量で、 $\alpha_k^\sigma = \frac{\pi_k^\sigma}{\pi_{k+1}^\sigma} \alpha_k$ ,  $\beta_k^\sigma = \left( \frac{\pi_k^\sigma}{\pi_{k+1}^\sigma} \right)^2 \beta_k$  である。したがって、残差ベクトル  $r_k$  をファイルに保存しておけば、 $\sigma$  だけ対角成分をシフトした連立方程式に対する  $\alpha_k^\sigma$ ,  $\beta_k^\sigma$  は  $\pi_k^\sigma$  から求めることができる。しかしながら、超大次元の対角化を考えると、この残差ベクトルの扱いに大きな問題が生じる。

本稿で考える超大次元計算の次元はおおむね 10 億次元以上であり、ベクトルは 1 本あたり数 GB から数十 GB と非常に大きい。そのため、各積分点毎に COCG 法が収束するまでの回数分、残差ベクトルを記憶するのは、記憶容量や I/O に要する時間が大きな障害となる。これを回避する方法は、ベクトルによる漸化式を適当な内積をとることでスカラー量の漸化式に変えることである。COCG 法の漸化式の  $x$  ベクトルは、収束すれば  $(z - H)^{-1}|\phi\rangle$  になるが、モーメント  $\mu_p$  としてはスカラー量である  $\langle \psi | \frac{1}{z - H} | \phi \rangle$  が必要となるだけである。 $|\psi\rangle$  に対応するベクトルを  $v$  と書くことにすると、モーメントの計算には内積  $vx$  の収束値がわかればよい。従って、ベクトル量に対する漸化式ではなく、 $v$  との内積をとったスカラー量に対する漸化式を考えればよい [9, 12, 13]。また、この手法とシフト法を使うと、すべての積分点に対する積分に必要な量が大量のベクトルをあらわに扱うことなく求められるので、COCG 法の収束過程でもエネルギーを計算することが超大次元計算に対しても現実的に可能になる。

収束過程においては、 $(z - H)^{-1}|\phi\rangle$  からずれ  $|\delta\rangle$  が COCG 法のステップ数とともに減少していくのだが、この減少の仕方は必ずしも単調ではなく、シェル模型の応用でも非連続的な振る舞いをすることが知られている [3]。本稿では、この振る舞いをステップ数に対するエネルギーの変化として理解し、その改良を考える。

このスカラー漸化式によって、エネルギーや波動関数の計算に含まれる Disk I/O は COCG 法の各ステップで、残差ベクトルをファイルに書くだけになる。また、エネルギー固有値だけでよく波動関数を求めないならば、Disk I/O はゼロにできる。ただし、

メモリーを節約する必要がある場合は、漸化式中の  $p$  をメモリーではなく、Disk 上に置く必要はある。いずれにしても、Disk I/O を、Lanczos 法や Thick Restart Lanczos 法に比べて SS 法では圧倒的に少なくできる。

### 3 大規模シェル模型計算

#### 3.1 計算に必要なメモリーの評価

一般に、量子多体系の超大次元の対角化をする場合、ハミルトニアン行列要素数が膨大になり、そのすべてを計算機のメモリー上に記憶できないため、行列要素は記憶せず、必要なときに計算することになる。その結果、超大次元計算の場合、メモリー上にあるのは、ベクトルだけになる。そういうコード上の実装を前提として、次元  $N$  の関数として必要なメモリーなどの諸量を概算することが、超大次元計算では重要である。

以下、シングルコアによる計算の場合のメモリー量を考察する。Lanczos 法の場合は、2本の Lanczos ベクトルを単精度でメモリーにおけば計算ができるため、必要となるメモリー量は  $8N$  バイトになる。一方、SS 法では、単純には複素数単精度ベクトルが3本必要になるので  $24N$  バイトになり、実際、論文 [3] では、シェルモデルに対して、そのように計算したため、対角化の限度は2～3億次元程度になった。しかしながら、積分路をとるときの  $\epsilon$  を実軸上にとることで COCG 法およびそのシフト計算をすべて実数で行うことができ、必要なメモリーを半分にできる。また、中間的なベクトルである  $p_k$  をメモリーではなくファイルにおくことで、Lanczos 法と同等の  $8N$  バイトにすることができる。Lanczos 法は、すべての Lanczos ベクトルを Disk に保存する必要があり、再直交化も必要になるので、繰り返し回数が増大すると Disk I/O の待ち時間が非常に大きくなるが、SS 法では I/O に必要な時間は非常に小さく、ステップ数によらず一定である。各方法でのメモリー量、Disk 利用量、I/O の量をまとめると表1のようになる。

#### 3.2 Lanczos 法による計算

ベンチマークとして使用する計算は、 $^{56}\text{Ni}$  の基底状態と  $J=0$  の第一、二励起状態を  $pf$  shell をモデル空間とし、GXPF1A[10] という相互作用で扱った場合とする。これは、十数年前、著者と共同研究者がモンテカルロ殻模型計算によって求めたもの [11] に相当するもので、その後の計算機環境の進歩によって、その厳密対角化は可能になっ

	memory	disk	Read I/O
Lanczos	8N	4kN	4k(k-1)N
Thick-restarted Lanczos	8N	4MN	4M(M-1)N[K/M]
SS	8N	4kN	0 or 4kN

表 1: 3つの方法に関して、必要なメモリー量、必要なディスク容量、および、I/Oに必要なデータ量を示した。 $k$ は収束までに必要なステップ数であり、 $M$ はリスタートする次元である。

た。しかしながら、その次元は *full pf* シェルの場合には  $M$  スキームで約 10 億となり、疎行列とはいえ、励起状態まで含めた場合には、この対角化はこれから示すように必ずしも容易なものではない。

図 3 では、各固有状態のエネルギーが Lanczos 法のステップ数とともにどのように変化するかを示した。Lanczos 法における初期ベクトルは  $4p4h$  空間での基底状態とした。基底状態は 10 ステップ程度で、第一励起状態は 20 ステップ程度で収束することがわかる。しかしながら、第二励起状態には 80 ステップ程度のステップ数が必要となり、途中で異なるエネルギーでいったん収束しそうになるという振る舞いがみられる。これは、Lanczos 法の初期ベクトルとして用いた  $4p4h$  状態の基底状態に、第二励起状態の成分がほとんど含まれていないためだと考えられる。

図 4 では、この計算に必要となった時間について、各ステップに要する時間を示した。この計算に使用したのは、i7-980 という 6 コアの CPU である。約 8 ステップまではおよそ 38 分であるが、それ以降は、徐々に増えていくことがわかる。計算時間の内訳は、主に、行列要素を生成するのにかかる時間と Disk I/O に要する時間である。約 8 ステップまでの計算時間は、この行列要素の計算時間がほとんどを占めているが、次第に I/O の時間が増えていく。このような計算時間の振る舞いは、Lanczos ベクトルの再直交化のためである。再直交化とは、Lanczos 法の漸化式を解いていく場合、Lanczos ベクトルの直交性が数値誤差のためにすぐに悪化するため、不可欠になる手順としてよく知られている。大次元での  $M$  スキームでのシェルモデルでは、この問題はいつそう深刻で再直交化を 2 度行う必要がある。従って、 $N$  ステップ目では、1 番目から  $(N-1)$  番目までの  $2(N-1)$  個分の Lanczos ベクトルを Disk から読み出す必要がある。約 10 ステップまではキャッシュメモリのおかげでその時間があらわにはならないと推定されるが、それ以降のステップでは計算時間がステップ数にほぼ比例しているのはこの I/O の時間のためと考えられる。また、全計算時間は、150.5 時間になるのに対して、行列要素の計算時間は約 70 時間であり、ほぼ半分以上の時間を I/O による時間が占めてし



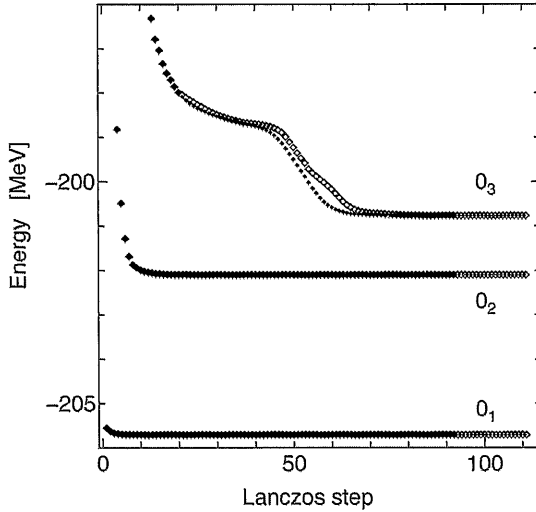


図 3:  $^{56}\text{Ni}$  の基底状態、 $0_2$ 、 $0_3$  の励起状態のエネルギーを Lanczos 法のステップ数の関数として示した。塗りつぶしたシンボルは普通の Lanczos 法の結果であり、白抜きのシンボルは Thick-Restart Lanczos 法の結果である。

まう。さらに Lanczos のステップ数が増大すれば I/O の時間はさらに大きくなるので、多くの励起状態を求めるのは極めて困難になる。

このひとつの解決法は、高速の記憶装置を導入することである。たとえば、大容量の SSD を RAID で接続することや大規模な並列機の分散メモリーを利用することが挙げられる。しかしながら、本稿ではアルゴリズムにより I/O を減らすことを考えることにする。

アルゴリズム的な解決として有効な方法は、Lanczos 法のリスタートを考えることである。 $M_0$  までステップが進んだときに、近似的な固有ベクトルをいったん求め、その近似固有ベクトルを再び Lanczos 法の初期ベクトルとして用いることである。こうすると、Lanczos 法に必要とされる Disk 容量が緩和され、I/O の時間も減少する。このリスタートは Lanczos 法を使うと誰もがすぐに考えることではあるが、実際に行うと  $M_0$  をあまり小さくしすぎると収束性が悪化することに気づく。さらに、複数の固有状態を求めるときには、アルゴリズム的な問題があることもすぐにわかる。後者の問題を解決したのが Thick Restart 法であり、Lanczos 法で生成される 3 重対角化行列をすこし拡張することになるが、最小限の拡張で済むアルゴリズムになっている。実際に、Thick Restart 法で計算した結果が図 3 の白抜きのシンボルで表したものである。

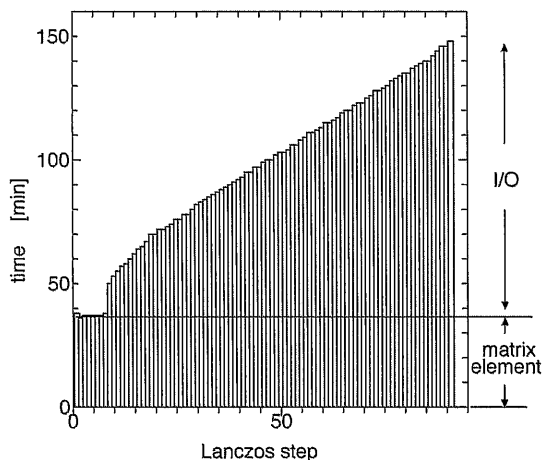


図 4:  $^{56}\text{Ni}$  の基底状態、 $0_2$ 、 $0_3$  の励起状態のエネルギーを求めるのに必要とされる Lanczos 法の計算時間について、Lanczos 法のステップ数の関数として一回の Lanczos 法のステップに必要な時間をヒストグラムで表示した。

ここでは、3つの固有状態を考え、 $M_0 = 20$  とした。基底状態、第一励起状態では 20 ステップ以下ではほぼ解けてしまうので、普通の Lanczos 法と同一の結果となる。Thick Restart 法の効果が表れるのは第二励起状態であり、やや収束が悪くなるが同一のエネルギーに収束し、うまく機能していることがわかる。この場合、I/O は  $M_0 = 20$  で制限されるので、Lanczos 法に比較すると I/O 時間は大幅に削減されるが、ゼロになるわけではない。

### 3.3 SS 法による計算

次に SS 法による計算について考えてみる。Lanczos 法の場合、励起状態を求める場合でも、固有値の一番小さいものから順に解いていくことになるが、SS 法の場合には、ターゲットしたエネルギーを直接求めてしまうことができる。この点は Lanczos 法と比較すると SS 法の明らかな利点である。また、ファイルの I/O 時間も大幅に緩和される。しかしながら、励起状態に対する COCG 法の収束性の問題は残る。ここでは、大次元計算時の SS 法の利点と課題について考えてみたい。

2.3 節でみたように、各積分点に必要な連立方程式をシフト COCG 法で解く際、漸化式をスカラー化する技法を使うと、連立方程式を解く際に容易にエネルギーの計算を

することができる。その結果、連立方程式を解く際にクリロフ部分空間の次元を増やすステップ数の関数として、エネルギーの収束性が評価できる。これまでは、この収束性は残差ベクトルのノルムで考えるのが普通であったが、本稿では、量子系を対象とするので、それをエネルギーに焼き直して考える。

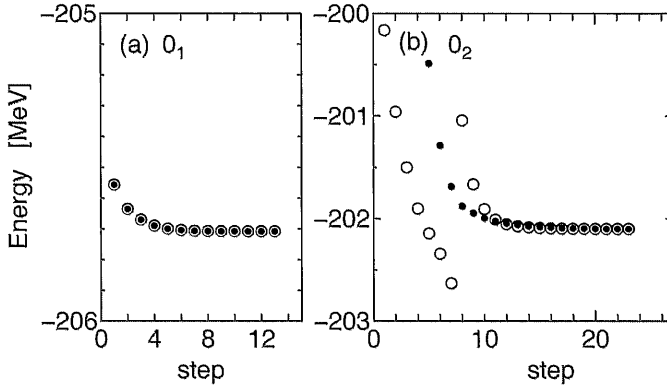


図 5:  $^{56}\text{Ni}$  の基底状態と  $0_2$  の励起状態のエネルギーをステップ数の関数として示した。Lanczos 法の結果は、塗りつぶしたシンボルで、SS 法の結果は、白抜きのシンボルである。

### 3.4 収束性の比較

まず基底状態の計算結果をみてみよう。図 5 (a) をみると基底状態の収束性について Lanczos 法と SS 法は同一の結果になることがわかる。このことは、Lanczos 法のステップ数と SS 法の連立方程式を解く COCG 法のステップ数は同一のクリロフ部分空間の次元に対応するからだと考えられる。また、COCG 法の収束過程における波動関数のエネルギーの変化は、基底状態に対しては変分が成り立つので、真の値を下回ることではなく、ステップ数に対するエネルギーの変化は比較的単調になる。

次に  $J = 0$  の第一励起状態の収束性を考える。図 5 (b) には、 $4p4h$  空間での基底状態を初期ベクトルとした Lanczos 法の収束性と  $4p4h$  空間での  $J = 0$  の第一励起状態を使った SS 法の収束性を示した。

励起状態に対しては、COCG 法の未収束の場合の解を用いると、基底状態とは異なり変分原理を満たさないため、エネルギーが真の値よりも下回ることが原理的に可能である。実際に図 5 (b) をみると、いったん第一励起エネルギーの真の値よりも下がって

しまうが、その後、再びエネルギーはあがり、徐々に正しい値に収束していく。COCG法の残差ベクトルの大きさをステップ数に関して収束の様子をみると、単調には収束せず、残差ベクトルのノルムがときどき非常に大きくなる過程を経ながらゼロになっていく。エネルギーのこの振る舞いはこの残差ベクトルのノルムの振る舞いと極めて似ている。また、収束するまでに要するステップ数は、Lanczos法でもSS法でもほぼ同じであるが、Lanczos法だと正しい固有値に上から収束していくが、SS法ではそれが保証されないので、収束の判断は難しくなるように思える。しかし、この点は、COCG法の残差ベクトルのノルムをみることで収束の判断はできる。また、得られた波動関数に対してエネルギー偏差を計算すれば固有状態かどうかは判断できる [3] ので、問題とはならない。

収束性を改善するには、SS法の初期ベクトルとして、より真の解に近いものを採用することである。1つは、トランケーションと組合せ、SS法で、ターゲットする励起状態をシェルモデル空間を大きくしながら追っていくことである。具体的には、SS法により、 $4p4h$  空間での  $J=0$  の第一励起状態を初期状態として  $6p6h$  空間の  $J=0$  の第一励起状態を解き、つぎにこの波動関数を初期ベクトルとして、 $8p8h$  空間の  $J=0$  の第一励起状態を解くという手順を繰り返していけばよい。その結果は、 $0_2$  と  $0_3$  の励起状態について、図6 (a,b) に示した。次元が小さい  $6p6h$  空間や  $8p8h$  空間の計算では収束性は単調でなく、かつ、それなりのステップ数が必要になるが、計算規模が大きくなる  $10p10h$ ,  $16p16h$  の計算に必要なステップ数は大幅に減少した。これは、より真の解に近い波動関数から計算をしたのだから当然の結果ではあるが、同様なことを Lanczos法で行うと、ターゲットした固有値だけを追うことはできないため、うまくいかない。これは励起状態をターゲットできるSS法の利点といえる。

もう一つの収束の改善法は、初期波動関数から *full pf* 空間においてすでに求めた基底状態の成分を引くことである。SS法を波動関数の改良という面からみていくと、複素関数論による計算で積分路の外側の波動関数の成分を切り捨てて波動関数を改良する部分と、COCG法でクリロフ部分空間で波動関数をよくしていくという、2つがあることがわかる。励起状態を考える時、COCG法のアルゴリズムでは、ターゲットした励起状態以外の波動関数の成分を取り除くには、反復を繰り返すのであるが、その収束性はやや特異で残差ベクトルのノルムをみるとかなり振動するようである。

量子系の研究では、基底状態から順々に励起状態を求めることが多いので、SS法でもエネルギーの低い方から解くことを考え、ある励起状態を求めたい場合は、それよりもエネルギーの低い状態との直交性を利用することを考える。具体的には、SS法にセットする波動関数  $|\phi\rangle$  を、ターゲットする状態よりもエネルギーの低いすでに求めた状態と直交化させておくのである。数値誤差を除けばクリロフ部分空間からもこの空間は排除されるので、原理的にはCOCG法の解からもそういう成分は除かれるはずである。その結果、COCG法の収束性はかなりよくなることが期待できる。また、SS法の

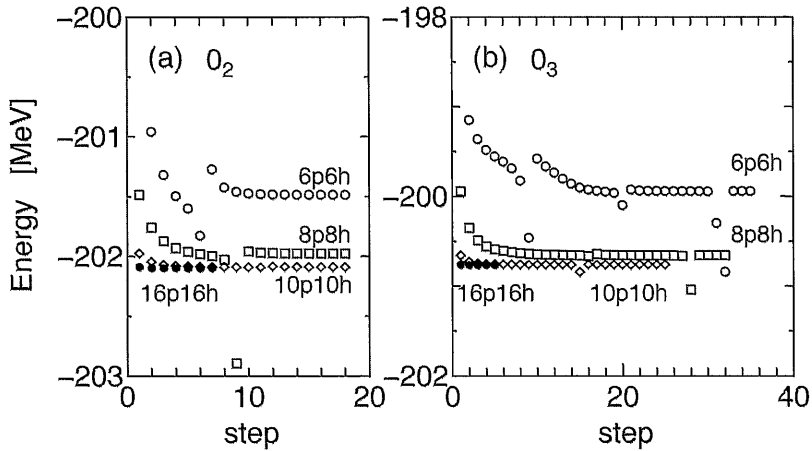


図 6: 6p6h, 8p8h, 10p10h, 16p16h の空間における  $^{56}\text{Ni}$  の  $0_2$  と  $0_3$  の励起状態のエネルギーの収束の様子を、SS 法のステップ数の関数として示した。

モーメント  $\mu_p$  の  $|\psi\rangle$  を  $|\psi\rangle = |\phi\rangle$  とおくことで、たとえ、数値誤差のせいで COCG 法の解の直交性が破れても、モーメントの計算からは完全に除かれるはずである。

実際に数値実験で、収束性の改善について確認してみよう。図 7 (a) は、4p4h 空間の  $0_2$  状態を使って 6p6h 空間における  $0_2$  状態を SS 法で解いた結果である。 $0_1$  状態と直交化をしない場合は、COCG 法の収束過程で、求めたいエネルギーよりも下回ったあと、大きく振動するという現象が現れるが、直交化しておけば、その変動が少なくなり収束性がよくなることが確認できる。図 7 (b) では、 $0_3$  状態に関して同じ計算を行った。この場合、すでに求めた  $0_1$  と  $0_2$  の状態と直交化させた。この場合も、同様に収束性が改善されることがわかる。

また、同様の直交化は Lanczos 法に対しても可能なので、どうなるかみておこう。図 7(a) の 6p6h 空間における  $0_2$  状態の Lanczos 法による計算では、同様に  $0_1$  状態と直交化した初期ベクトルから始めると、SS 法の結果とほとんど同じになる。しかしながら、図 7(b) で示すように  $0_3$  状態に関しては微妙であり、途中までは Lanczos 法も同じ結果を与えるが、直交化の精度が悪化し、よりエネルギーの低い異なる解のほうに進んでしまう。その分岐点は SS 法でも微妙な変化のある点であるのは興味深い。本稿では計算はしていないが、Lanczos 法の場合は、初期ベクトルだけではなく、すべての Lanczos ベクトルに関して直交化の条件を課すか、または、ハミルトニアン  $H$  を  $H + \alpha \sum_i |i\rangle\langle i|$  と変更し、数値誤差で発生する、考察している状態よりもエネルギーの低い固有状態の成分にペナルティーを与えるようにすれば改善できるかもしれない。ただし、 $|i\rangle$  は固

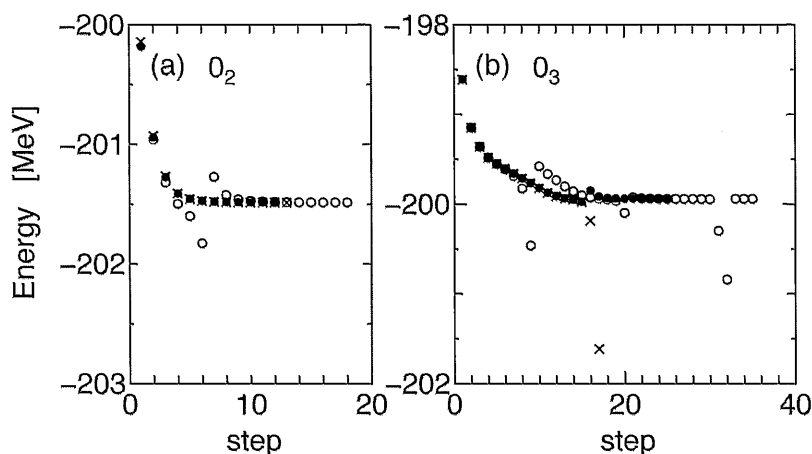


図 7:  $6p6h$  空間における  $^{56}\text{Ni}$  の  $0_2$  と  $0_3$  の励起状態のエネルギーを SS 法のステップ数の関数として示した。図 (a) の白抜きのシンボルは  $4p4h$  空間の  $0_2$  状態を使って SS 法で解いた結果である。それに対して、黒塗りのシンボルは、 $4p4h$  空間の  $0_2$  状態から  $6p6h$  空間の  $0_1$  の成分を抜いた状態を使って SS 法で解いた結果である。図 (b) の白抜きのシンボルは、 $4p4h$  空間の  $0_3$  状態を使って SS 法で解いた結果である。それに対して、黒塗りのシンボルは、 $4p4h$  空間の  $0_3$  状態から  $6p6h$  空間の  $0_1$  および  $0_2$  の成分を抜いた状態を使って SS 法で解いた結果である。×は同様に直交化した初期ベクトルから計算した Lanczos 法の結果である。

有状態で、 $i$  は考察している状態よりもエネルギーの低い固有状態に対して和を取る。 $a$  は適当な正の数である。

図 8 は、 $full\ pf$  空間に対して、直交化したベクトルによって  $0_2$ 、 $0_3$  状態を SS 法により解いたものである。図 8(b) では  $4p4h$  空間の  $0_3$  状態を使って  $16p16h$  空間で解いた結果、図 3 にくらべて収束が速まり、かつ、この領域に 3 本の固有状態が求まった。

本稿では、COCG 法の収束過程でも SS 法の複素積分を実行して、収束過程をエネルギーで考えるという方法を導入した。その動機は、残差ベクトルのノルムだとどういふ状態の成分が混ざっているかがわかりにくいからである。求めたい状態の真のエネルギーよりも下がったということは、その状態よりもエネルギーの低い状態が混ざったせいであるから、それをあらかじめ取り除くことでエネルギーが改善するのは、量子力学では至極当然の結果といえる。一般に COCG 法の収束性は固有値の集積度合に依存す

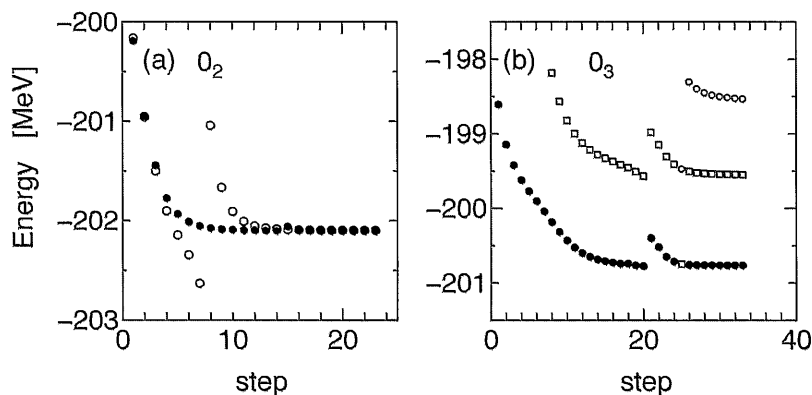


図 8: 図 7 で行った SS 法による計算を *full pf* 空間でおこなったもの。図 (a) の白抜きシンボルは直交化なしの SS 法の結果であり、黒塗りのシンボルは直交化した SS 法の結果である。図 (b) では直交化した SS 法の結果のみを示した。この初期状態からは 3 本の励起状態が求まる。

ることが知られている。特に、固有値が図 9 のように固有値列の端に位置すると収束性はよくなる。量子系では、基底状態がそれにあたる。ここで起こった励起状態の収束性の改良の方法は、直交化によりターゲットした固有値を端にするという操作にあたるといえる。

## 4 まとめ

本稿は、SS 法の超大次元対角化への応用の現状について Lanczos 法と比較しながらまとめたものである。SS 法は SAKURAI と SUGIURA が 2003 年に提案したコーシー積分を利用した全く新しい対角化法 [2] である。その量子多体系への応用 [3] はすでに発表した。本稿はそれ以降の発展を 10 億次元を超える超大次元の場合についてその可能性と課題をまとめた。

超大次元の場合は、計算時間に加え、必要なベクトルを Disk 等に記憶して利用するための I/O の負担が非常に大きいことが知られている。たとえば、Lanczos 法の場合は、Lanczos ベクトルの記憶容量が莫大となり、かつ、I/O を頻繁に行う必要が生じるため、その I/O の負担は大きく、特に励起状態の計算がきわめて困難になることが知られている。今回、その改善が SS 法によってできることがわかり、さらに、その収束

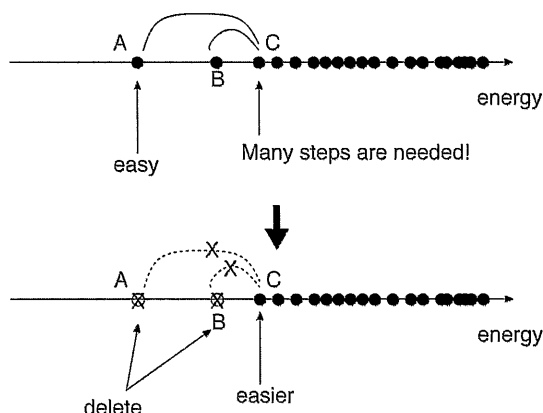


図 9: 固有値の密集度と収束性：上の図のように固有値列の端にある固有値Aの場合は収束が速いが、固有値Cは固有値Aよりも収束は遅くなる。下の図のように、固有値A、Bを削除すれば、固有値Cは固有値列の端になるので、収束性があがる。

性を上げるためのテクニックを見出した。

励起状態の波動関数を求めるためには、SS法においても、Lanczos法と同様に、エネルギーの低い状態から順次解いていき、SS法の初期ベクトルから、求めたい状態よりもエネルギーの低い状態を除いてしまうと収束性があがることがわかった。また、トランケーションを利用して励起状態の近似波動関数を作ることも非常に有効であることがわかった。

ここでは、議論しなかったが、Lanczos法と同様にSS法においてもリスタート法を加えることができる。COCG法の収束判定を緩めた段階でSS法の解を求め、再び、その解をSS法の初期ベクトルにすることができる。この拡張は非常に簡単である。また、Lanczos法においても直交化した初期状態を使った計算ができるはずで、うまくいく場合は、SS法と同等の収束性を示す。これは、背景にあるクリロフ部分空間が一致するからだと考えられる。しかし、Lanczos法のもつ数値的不安定性から、直交性が破れてしまうことがある。その場合は、Lanczos法の各ステップで直交条件を課すといわれるが、仮にうまくいっても、SS法よりもI/Oが増えてしまうであろう。

残された課題は、SS法における高励起状態の計算である。複素積分によりどのエネルギーの状態でも求められるわけだが、COCG法の収束回数は、エネルギーがあがるほど増えてしまう。これは今後の課題としたい。

本研究は、平成23年度専修大学研究助成「SS法による大規模シェル模型計算とそ



の並列化」によって行われた。筑波大の櫻井氏には、SS法について詳しく教えていただき、さらに、シュル模型への応用に関して共同研究をしていただいたこと、九産大の金子氏には長年の共同研究に、また、東大の清水氏には OpenMP による並列化および Thick Restart Lanczos 法の実装について、感謝したい。

## 参考文献

- [1] C. Lanczos, J. Res. Nat. Bur. Stand, **45**, 255 (1950).
- [2] T. Sakurai, H. Sugiura, J Comput Appl Math 2003, 159, 119, T. Ikegami, T. Sakurai, U. Nagashima, Technical Report CS-TR-08-13, Tsukuba, 2008.
- [3] T. Mizusaki, K. Kaneko, M. Honma, and T. Sakurai, Phys. Rev. C **82**, 024310 (2010).
- [4] K. Wu and H. Simon, SIAM Journal on Matrix Analysis and Applications, v 22, No. 2, pp. 602-616, (2001).
- [5] R. R. Whitehead and A. Watt, J. Phys. G: Nucl. Phys. **4**, 835 (1978).
- [6] H. A. van der Vorst and J. B. M. Melissen, IEEE Trans. on Magnetics **26**, 706 (1990).
- [7] R. Takayama, T. Hoshi, T. Sogabe, S.-L. Zhang, T. Fujiwara, Phys. Rev. B **73**, 165108 (2006).
- [8] H. Ohno, Y. Kuramashi, H. Tadano, T. Sakurai, JSIAM Letters Vol. 2, pp.115-118 (2010).
- [9] S. Yamamoto, T. Sogabe, T. Hoshi, S. -L. Zhang and T. Fujiwara, J. Phys. Soc. Jpn, 77 (2008), 114713.
- [10] M. Honma, T. Otsuka, B. A. Brown, and T. Mizusaki, Eur. Phys. J. A 25 Suppl. **1**, 499 (2005).
- [11] T. Mizusaki et al., Phys. Rev. C 59, R1846 (1999). この論文では、本稿とは異なる相互作用を使っているが、計算規模の面では同一とみなしてよい。
- [12] Y. Futamura, H. Tadano, and T. Sakurai, JSIAM Letters, Vol. 2, pp. 127-130, (2010).
- [13] Y. Futamura, and T. Sakurai, private communication.

